Position based routing in crowd sensing vehicular networks

Published Version:
Position based routing in crowd sensing vehicular networks / Bazzi A.; Zanella A.. - In: AD HOC NETWORKS. - ISSN 1570-8705. - ELETTRONICO. - 36:2(2016), pp. S1570870515001304.409-S1570870515001304.424. [10.1016/j.adhoc.2015.06.005]

This version is available at: https://hdl.handle.net/11585/713193 since: 2020-01-12

Published:

DOI: http://doi.org/10.1016/j.adhoc.2015.06.005

*Alessandro Bazzi, Alberto Zanella, **Position based routing in crowd sensing vehicular networks**, Ad Hoc Networks, Volume 36, Part 2, 2016, Pages 409-424*
ISSN 1570-8705

The final published version is available online at:
https://doi.org/10.1016/j.adhoc.2015.06.005

Position Based Routing in Crowd Sensing Vehicular Networks

Alessandro Bazzi*, Alberto Zanella

CNR/IEIIT, Viale Risorgimento 2, 40136 Bologna, ITALY

Abstract

Using vehicles as sensors allows to collect high amount of information on large areas without the need to deploy extensive infrastructures. Although cellular technologies are presently the only solution to upload data from vehicles to control centers, in the next future short range wireless technologies could be used to offload part of this data traffic through vehicle to vehicle and vehicle to roadside communications. In such scenario, the greedy forwarding (GF) position based routing is an interesting algorithm to efficiently route packets from vehicles to the destination. However, GF suffers from the well known problem of local minima, which causes part of the packets to remain blocked in certain areas of the scenario. To deal with this issue, we propose two novel routing algorithms, specifically designed for crowd sensing vehicular networks (CSVNs): GF with available relays (GFAVR), fully distributed and independent of the scenario, and GF with virtual roadside units (GFVIR), exploiting a preliminary design phase where local minima are located. Through extensive simulations performed in different realistic urban scenarios, results demonstrate that both algorithms allow to improve data delivery by 10 to 40%, with negligible overhead and limited increase of complexity.

Keywords: Crowd sensing vehicular network; Position based routing; VANET; IEEE 802.11p.

1. Introduction

Short range vehicular communications will enable in the next years the paradigm of connected vehicles. In August 2014, the National Highway Traffic Safety Administration (NHTSA), one of the main USA agencies in the field of transportation, issues an Advance Notice to proceed with standardization of vehicle to vehicle communication for light vehicles [2] and similar decisions will probably be taken by institutions of other Countries. It is thus expected that new vehicles will be soon equipped with wireless short range communication systems such as the wireless access in vehicular environment (WAVE)/IEEE 802.11p technology [3].

Even if this technology is primarily foreseen for safety purposes, other applications could take benefit from its deployment and the consequent creation of vehicular ad hoc networks (VANETs). In particular, short range multi-hop communications could be used to offload cellular networks, that are challenging an increasing bandwidth request; crowd sensing vehicular network (CSVN) applications are among the main specific applications where cellular offloading could be performed effectively [4]. Crowd sensing is an emerging paradigm that takes advantage of pervasive mobile devices (such as smartphones or in vehicle sensors) to efficiently collect data, enabling numerous large scale applications [5, 6]. Focusing on vehicular scenarios, some million vehicles are today equipped with on board units (OBUs) that periodically collect information from various sensors to be sent to a remote control cen-

☆This manuscript has been accepted for publication in Ad Hoc Networks. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all disclaimers that apply to the journal apply to this manuscript. A definitive version will be subsequently published, DOI: 10.1016/j.adhoc.2015.06.005
This work was supported in part by the CNR project ICT for smart mobility (ICT4SM). Parts of this work were presented in [1].

*Corresponding author
Email addresses: alessandro.bazzi@cnr.it (Alessandro Bazzi), alberto.zanella@cnr.it (Alberto Zanella)

ter. Presently, they are used for insurance purposes and traffic estimations, but other applications have been proposed, like urban environment surveillance [7] or widespread pollution measurements [8]. For the moment, only cellular networks are used to upload data from the OBUs, with high costs in terms of billing and a large impact on cellular resource usage [9]. However, in the near future, short range roadside units (RSUs) are expected to be deployed in cities and highways to help collecting data from the vehicles.

Dealing with the use of short range technologies in CSVNs, the main issues to maximize the performed offloading are surely the RSU placement and the design of routing protocols [10, 11, 12, 13]. As clarified in the further, even if several routing algorithms have been proposed for VANETs, in most cases they do not deal with the peculiarities of CSVNs or they are too complex for a large scale implementation. One protocol which represents a simple yet effective solution for CSVNs is greedy forwarding (GF), which foresees that each OBU selects as next hop the neighboring OBU which maximally reduces the distance from the nearest RSU [14]. This protocol, however, suffers from the well known problem of local minima (or local optima), that causes packets to be collected in specific areas of the road network and never delivered to the RSUs [15, 16]. This effect, the implications of which are further described in the paper, can be reduced by optimally placing RSUs, as suggested for example in [10, 11]. However, first this approach faces the constrains on site availability, which is not always guaranteed, and second it cannot eliminate the problem in all scenarios.

To deal with the local minima problem, even in the presence of non optimal RSU placement, we propose two novel routing algorithms that are specifically designed for CSVNs. The first algorithm, denoted GF with available relays (GFAVR), is fully distributed, and foresees that each vehicle estimates its own positioning in a local minimum. The second algorithm, denoted GF with virtual RSUs (GFVIR), exploits a preliminary design phase where local minima are estimated and alternative routes are identified.

The effectiveness of both algorithms is shown through extensive simulations performed in two urban scenarios, characterized by different sizes and different vehicle densities: the city of Bologna (Italy) and the city of Cologne (Germany).

The paper is organized as follows: The related work is discussed in Section 2; In Section 3, the system model and the addressed problem are defined; Section 4 focuses on GF and the problem of local minima; The two proposed algorithms, GFAVR and GFVIR, are then detailed in Sections 5 and 6, respectively; The assumptions made and the simulation settings are shown in Section 7 and results are provided in Section 8; Finally, our conclusion is given in Section 9.

2. Crowd sensing vehicular networks and related work

Due to the wide diffusion of consumer devices with sensing abilities, such as smartphones and media players, their use to obtain large scale information from the environment (crowd sensing) has recently drawn a large interest from researchers and industries [5, 6].

This paradigm has been also investigated in the vehicular scenario adopting several other names, including vehicular sensor networks (VSNs) (e.g., in [17]), probe vehicles (e.g., in [18]), or floating car data (FCD) (e.g., in [19]). An interesting survey on this topic can be found in [20]. Among the example applications that have been envisioned we can cite the improvement of urban environment surveillance [7], the provision of large scale pollution measurements [8], the alerting of upcoming vehicles when an accident is observed [21], and the enabling of traffic monitoring [22]. Besides possible applications, many other aspects have been investigated, like the data management at the control center [23] and the aggregation of messages among neighbor vehicles to reduce the amount of information sent to the control center [24].

CSVNs can be seen as the intersection of wireless sensor networks (WSNs) and VANETs; their peculiarities are [25]:

- Nodes collect information to be delivered to a control center (like in WSNs);
- The high mobility makes the node density and the network topology changing frequently (like in VANETs).

To collect the information from OBUs, CSVNs can rely on either cellular networks or short range communications. In the latter case, the overall architecture must be completed with the placement of RSUs, connected to the control center, and one of the main challenging aspects is the definition of the routing protocol that allows data to reach these RSUs. Several routing protocols have been proposed for VANETs in the last years, including those described in [12, 13, 26]. Some of the proposed algorithms, including as an example CAR [27], are reactive, i.e., they search for a path towards a destination only when a packet to that destination is enqueued. This approach is normally preferable in slowly variable ad hoc networks, since it minimizes the signaling overhead; however, the main drawbacks are that i) it needs a search phase to define the route, which might be a problem in the quickly variable vehicular scenario, and ii) it suffers from scalability problems in large networks [28]. For these reasons, and based on the possibility to send periodic messages for safety purposes (denoted as beaconing in the further), most protocols are proactive, i.e., they continuously update a table towards the possible destinations, independently from the presence of packets to that destination in the transmission queue. Examples are greedy perimeter stateless routing (GPSR) [15] and Greedy Perimeter Coordinator Routing (GPCR) [16]. Some protocols, such as EPIDEMIC [29] or SPRAY&WAIT [30], also foresee the use of multiple copies. Allowing multiple copies of a packet, however, has the drawback that no OBU carrying one of the copies knows whether the other copies have been already delivered or not, increasing, in general, the network load. Finally, several algorithms rely on additional and detailed (thus costly) information that must be carried by OBUs, such as road maps (e.g., GeoSVR [31]), traffic signal schedule (e.g., ROAMER [32]), information on buses and their routes (e.g., SKVR [33]), or the routes that are daily traveled by vehicles (e.g., PER [34]).

Although most protocols designed for vehicular networks can be also applied to CVSNs, only few proposals have been explicitly designed for a CVSN scenario, characterized by the fact that the position of the destination (one of the RSUs) is fixed and known by vehicles [25]. To this regard, an algorithm that perfectly suites to this scenario is the GF, which also has other useful properties, as detailed in the next Section. Unfortunately, the presence of local minima tends to decrease the performance of such algorithm [15, 16], as deepened in Section 4. To overcome this important limitation, we propose and investigate the performance of the two novel routing protocols that are explicitly designed for CVSN scenarios.

3. System model and problem definition

Definitions. Hereafter, we use vehicle-to-cellular (V2C) to denote communications involving the cellular connection of an OBU, vehicle-to-roadside (V2R) to denote communications between an OBU and an RSU, and vehicle-to-vehicle (V2V) to denote communications between OBUs.

Application. Although various applications could be considered, we focus as an example case to the collection of information for insurance purposes. We thus assume the following.

- Data cannot be modified, thus filtering or aggregation (such as in [24]) cannot be performed during the delivery phase;
- Data management and long term storage are left to the remote control center;
- Each packet must be delivered to the control center, thus packets that do not reach an RSU must be sent using V2C.

Although other applications might have less stringent requirements, relaxing the first or the second one would only reduce the amount of data to be delivered to the control center and would not limit the validity of the routing protocol comparison we provide. Relaxing the third would cause localized loss of data (as also demonstrated in Section 8), which is undesirable for any CSVN application.

On board units. We assume that all vehicles are equipped with an OBU that periodically collects data from sensors to be delivered to a remote control center. All OBUs are assumed equipped with a positioning system such as the global positioning system (GPS), a cellular technology, and the short range wireless technology detailed in the further. RSUs, equipped with the same short range technology, are deployed to collect packets from vehicles and forward them to the control center through a high speed link.

To maximally offload cellular networks, OBUs will use V2R anytime they are connected to an RSU. Otherwise, a routing algorithm is adopted to find the best route towards an RSU through multiple V2V hops. In particular, the routing algorithm is in charge to find the next relay among the neighbor nodes. Neighbor nodes are those nodes to which the OBU is connected; they are known thanks to a beaconing service, that is, through messages that are periodically broadcasted by all OBUs to advertise their position, direction, and other metrics used for safety purposes.

To avoid packet losses, whenever the number of packets inside the transmission buffer of an OBU reaches a given threshold, the OBU sends part of them through V2C. We also assume a maximum tolerated delay for the message delivery. In particular, each message carries a timestamp of the instant of generation; focusing on the oldest message in the queue, when the difference between the current and the generation time exceeds a given threshold, all messages in the queue are sent through V2C.

Routing. Concerning the routing algorithm, the peculiar aspects of CSVNs are: i) the transmissions are performed from the OBUs to the RSUs, and ii) mobility makes the topology frequently changing.

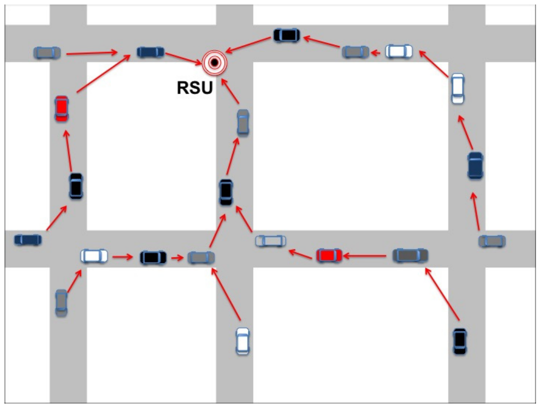
As already discussed, data loss is not acceptable, thus only unicast transmissions with MAC level acknowledgments are possible, and only single copy routing is considered. Under such conditions, proactive routing tends to be preferred for the reasons detailed in Section 2 and the use of maps, with the related updating issues and costs, is avoided.

These guidelines, discussed more in deep in [25], exclude most of widely considered routing algorithms for VANETs. For example, CAR [27] is not suitable since it is reactive, GSR [35] because it requires maps on board, and SPRAY&WAIT [30] due to the use of multiple copies.

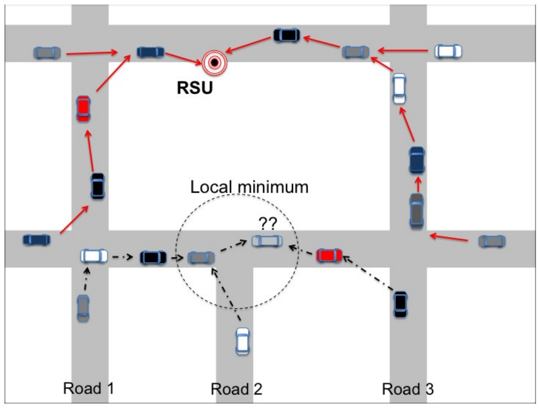
Among those that respect all the listed requirements, a simple yet effective solution is GF.

4. Greedy forwarding and the local minima in crowd sensing vehicular networks

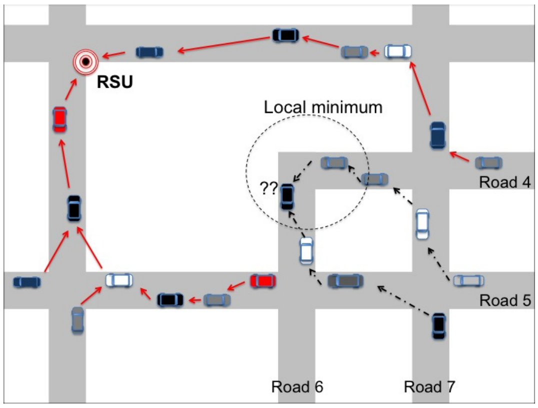
GF works as follows. Each OBU knows its own position by the positioning system, the position of its neighbors thanks to the beaconing service, and the position of RSUs provided by a location service. Although the location service definition is out of the scope of the present work, an example could be the provision of a small database to be occasionally updated with new deployments or other changes, on a periodical basis; a wider discussion of location services can be found for example in [36, 37]. With this information, each OBU that is not directly connected to an RSU first selects the nearest RSU as the destination, then chooses as next relay the neighbor that maximally reduces the distance to that RSU. As long as there are no neighbors closer to the destination, the packets are stored and carried. An example of GF behavior is shown in Fig. 1(a).



(a) Greedy forwarding.



(b) Local minima. Example 1.



(c) Local minima. Example 2.

Figure 1: Greedy forwarding and local minima. Red solid arrows are used for the connections that will finally reach the RSU. Black dash-dotted arrows are used for the connections that bring to a local minimum area.

Table 1: Notations used in the GFAVR and GFVIR descriptions.

Used in	Symbol	Meaning
Both algorithms	Θ_k	Generic OBU k
	$\mathcal{N}_{\Theta_k}$	Set of neighbors of OBU k
	R_{Θ_k}	Nearest RSU to OBU k
	T_B	Time interval between two beacon generations
	$\mathcal{N}^*_{\Theta_k}$	Set of neighbors that are relay available and closer than Θ_k to R_{Θ_k}
GFAVR	$\underline{d}$	Minimum distance for the angle check
	$\mathcal{N}^f_{\Theta_k}$	Set of neighbors of OBU k farther than $\underline{d}$
	α_{max}	Largest angle that neighbors of OBU k, taken two by two, form using OBU k as vertex
	$\underline{\varphi}$	Maximum angle to consider all neighbors in the same direction
GFVIR	$V_j^{(A)}$	Generic AVRSU j
	$V_j^{(S)}$	Generic SVRSU j
	$\rho_{V_j^{(A)}}$	Exclusion distance of AVRSU j
	$\rho_{V_j^{(S)}}$	Exclusion distance of SVRSU j
	$\mathcal{V}_x^{(A)}$	Set of AVRSUs referred to RSU R_x
	$\mathcal{V}_x^{(S)}$	Set of SVRSUs referred to RSU R_x

GF suffers, however, from the local minima problem: if the source node is nearer to the addressed destination than all its neighbors (and the destination is out of the node's coverage), then the destination cannot be reached and the node is said to be in a local minimum. In vehicular scenarios this event occurs when the road layout is characterized by the presence of an area that is closer to the RSU of interest than all accessible areas in its proximity. This event is clarified through the two examples shown in Figs. 1(b) and 1(c), where the RSU is deployed in a position that causes a local minimum. With reference to Fig. 1(b), data generated by OBUs on Roads 1, 2, and 3, tend to be routed toward the local minimum region; the same happens in Fig. 1(c) for data generated by OBUs on Roads 4, 5, 6, and 7. The vehicle movements will only cause a modification of which OBUs are in the local minimum, continuously collecting data from the neighborhood without any possibility to reach the RSU.

Previous work tried to react to local minima through a procedure denoted *recovery strategy*, which is invoked anytime an OBU has no next hop towards the destination; the most cited algorithms providing a recovery strategy are GPSR [15] and GPCR [16]. However, GPSR has not been designed for high mobility scenarios and often fails in VANETs, with a significant increase of the number of transmissions and without a higher delivery rate [28]. GPCR improves GPSR by introducing the concept of junction nodes (i.e., OBUs that are positioned at junctions), but it is still problematic in real urban scenarios, mainly for two reasons [38]: first, the identification of a junction has high failure probability in GPCR; second, often the use of nodes at junctions is not needed or even counterproductive, since most junctions are not in a local minimum.

Since in CSVNs destinations are fixed and delay is tolerated, instead of implementing a recovery strategy, either traffic flows could be forced through directions that avoid the local minima or OBUs could store and carry packets when they are located inside local minima. Based on these considerations, two routing protocols are hereafter proposed, one fully distributed and the other based on a shared database. Whereas the former scheme is simpler, the latter provides better results in most cases, at the cost of a preliminary phase performed offline and customized to the specific scenario.

5. Distributed Approach: Greedy Forwarding with Available Relays

The first proposed algorithm, GFAVR, does not require any preliminary phase and is fully distributed. Each OBU acts autonomously, based on the local information. If the OBU is not covered by an RSU and does not have a neighbor available as next relay, the algorithm estimates if the vehicle is in a local minimum (as detailed in the further). If the algorithm assumes it is in a local minimum, the OBU broadcasts its own unavailability to act as a relay and neighbors avoid to consider it as a possible next relay.

5.1. Relay availability

Each OBU is assumed to be relay available when it is located out of a local minimum. More specifically, denoting with Θ_k the generic OBU, the relay availability is defined as follows.

Definition 1. ***GFAVR relay availability.*** Θ_k *is (GFAVR) relay available if ANY of the following conditions is fulfilled:*

1. *It is directly covered by an RSU;*
2. *It has a next relay selected towards the nearest RSU;*
3. *All its neighbors are aligned on the same road and Θ_k is located in one of the two extremities.*

An OBU which is not relay available is said *relay unavailable.*

The first two conditions state that, if an OBU can identify a next hop (either RSU or another OBU) for its stored messages, it is surely out of a local minimum area. The third condition in the definition is required to avoid that a vehicle marks itself as unavailable only because it does not have any neighbor in the direction of the RSU. In particular, the third one is added in the case the OBU cannot select a next hop node, and allows to distinguish between the following (opposite) situations:
1) The OBU either has no neighbor or all its neighbors are on the same road, although all located in the opposite direction with respect to the addressed RSU; this latter case is represented by OBU_1 in Fig. 2. This condition does not necessarily lead to a local minimum and the OBU is considered relay available.
2) The OBU has neighbors located in different directions, but none of these directions lead to the addressed RSU; this is the case of OBU_2 in Fig. 2. Under such condition, there is no way to get closer to the RSU and the OBU is probably in a local minimum. Under such condition, the OBU is considered as relay unavailable.

To determine whether all neighbors are on the same road and in the same direction or not, the generic OBU Θ_k exploits the (known) coordinates of the neighbors. Firstly, it excludes from the evaluations those neighbors that are too close, i.e. that are distant less than a given threshold $\underline{d}$; the rationale is that a vehicle on a different lane might otherwise be erroneously placed on a different road segment. Secondly, it checks the convex angle that the remaining neighbors create two by two, using Θ_k as the vertex, and compare them to a given threshold $\underline{\varphi}$. If two neighbors form an angle which is larger than the threshold $\underline{\varphi}$, Θ_k assumes there are neighbors located on different directions (see, for example, OBU_2 and its neighbors in Fig. 2).

The two described steps can be formalized as follows. Denoting with $d(A, B)$ the distance between A and B, with $\angle(A, B, C)$ the convex angle created by the two segments $\overline{AB}$ and $\overline{BC}$, and with $\mathcal{N}_{\Theta_k}$ the set of neighbors of OBU Θ_k, the first step is the evaluation of set $\mathcal{N}^f_{\Theta_k}$, as follows.

$$\mathcal{N}^f_{\Theta_k} = \{N_i \in \mathcal{N}_{\Theta_k} : d(N_i, \Theta_k) > \underline{d}\}. \tag{1}$$

$\mathcal{N}^f_{\Theta_k}$ excludes those neighbors that might be simply on different lanes. The second step is to evaluate the largest angle, as follows.

$$\alpha_{max} = max\{\angle(N_i, \Theta_k, N_j)\ \forall N_i, N_j \in \mathcal{N}^f_{\Theta_k}\}. \tag{2}$$

Finally, the OBU is relay available if $\alpha_{max} < \underline{\varphi}$; in such case, in fact, the OBU assumes not being in a local minimum, but simply having no next hop due to low vehicular density.

Note that, when an OBU is relay unavailable, it cannot be selected as next relay by neighbors; the nearest neighbors will then be unable to find a suitable next relay and become, in turn, relay unavailable. Thus, the relay unavailability will propagate to the neighboring vehicles until junctions are reached (or unconnected OBUs are present). Therefore, the propagation is confined in a limited area around the local minimum and does not propagate in other parts of the scenario.

An example of relay availability and relay unavailability is shown in Fig. 2. OBU_1 has two neighbors with an angle smaller than the threshold; it means that all neighbors are in the same direction and OBU_1 marks itself as relay available. On the opposite, OBU_2 has neighbors with an angle higher than the threshold, meaning that it is placed in a local minimum; OBU_2 marks itself as relay unavailable, and this will propagate to its neighbors. In our implementation, $\underline{d} = 20$ m and $\underline{\varphi} = \pi/8$ are used, according to the average road width in the considered scenarios.

5.2. The GFAVR protocol

Each OBU sends the relay availability in a single bit added to the beacon frame, every T_B, assumed the same for all OBUs for simplicity.

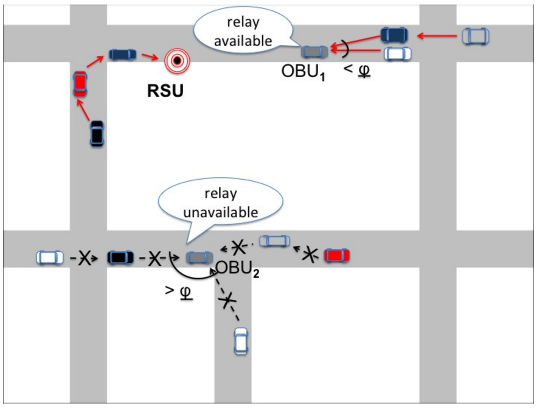


Figure 2: GFAVR. Examples of relay availability. Black dashed arrows with a black 'x' represent transmissions that are not performed due to relay unavailability.

Algorithm 1 GF with Available Relays

```
1: procedure BY OBU $\Theta_k$
2:
3: $\mathcal{R}$: set of RSUs
4: $R_{\Theta_k}$: nearest RSU to OBU $\Theta_k$
5: $\mathcal{N}_{\Theta_k}$: set of neighbors of OBU $\Theta_k$
6: $H_{\Theta_k}$: next hop for OBU $\Theta_k$
7: $\omega_X$: relay availability of OBU $X$
8: $T_B$: beacon interval
9:
10: Every $T_B$:
11:
12:     // Reset the relay availability
13:     $\omega_{\Theta_k}$ := true
14:
15:     // The nearest RSU is selected
16:     $R_{\Theta_k} := argmin_{R_r \in \mathcal{R}}\{d(\Theta_k, R_r)\}$
17:
18:     // Check if the nearest RSU provides coverage
19:     if $\Theta_k$ is connected to $R_{\Theta_k}$ then
20:         $H_{\Theta_k} := R_{\Theta_k}$
21:     else
22:         // The next hop is searched among neighbors
23:         $H_{\Theta_k}$ := null // Reset the next hop
24:         $d_{min} := d(\Theta_k, R_{\Theta_k})$ // Reset the min. distance
25:         for all $N_w \in \mathcal{N}_{\Theta_k} : \omega_{N_w} = true$ do
26:             if $d(N_w, R_{\Theta_k}) < d_{min}$ then
27:                 $H_{\Theta_k} := N_w$
28:                 $d_{min} := d(N_w, R_{\Theta_k})$
29:         // Relay availability is checked if no next hop
30:         if $H_{\Theta_k}$ = null then
31:             for all $N_x \in \mathcal{N}_{\Theta_k}$: $d(N_x, \Theta_k) > \underline{d}$ do
32:                 for all $N_y \in \mathcal{N}_{\Theta_k} - \{N_x\}$: $d(N_y, \Theta_k) > \underline{d}$ do
33:                     if $\angle(N_x, \Theta_k, N_y) > \underline{\varphi}$ then
34:                         $\omega_{\Theta_k}$ := false
35:                         break
36:
37:     // If $H_{\Theta_k} \neq$ null the next hop is addressed
38:     if $H_{\Theta_k} \neq$ null then
39:         Transmit data to $H_{\Theta_k}$ in the service channel
40:     else
41:         Store and carry data
42:
43:     // In any case, send the beacon
44:     Send beacon with $\omega_{\Theta_k}$ in the control channel
```

Each OBU Θ_k which is not covered by an RSU performs the following algorithm to select the next hop.

1. Θ_k finds the nearest (in the Euclidean sense) RSU R_{Θ_k};
2. Θ_k defines the set $\mathcal{N}^*_{\Theta_k}$ of the neighbor OBUs that are relay available AND closer than Θ_k to R_{Θ_k};
3. If $\mathcal{N}^*_{\Theta_k}$ is empty, then no OBU is selected as next relay by Θ_k. Otherwise, Θ_k selects as next relay the OBU in $\mathcal{N}^*_{\Theta_k}$ which is the closest to R_{Θ_k}.

To follow possible variations in the topology, in our implementation we assume all vehicles repeat the algorithm before sending their beacon frame, which occurs every T_B seconds.

A pseudo code description of the algorithm is shown in Algorithm 1.

5.3. Complexity of GFAVR

Compared to GF, the GFAVR protocol implies the addition of a single bit in the beacon messages and a very small increase of complexity in the routing protocol performed by each OBU. More specifically, with Θ_k denoting the generic OBU, $\mathcal{N}_{\Theta_k}$ the set of neighbors of Θ_k, and $\#\mathcal{X}$ the cardinality of set $\mathcal{X}$, the following additional elements are required.

- One signaling bit is added in each beacon message sent by Θ_k to advertise if Θ_k is relay available or not.

- Periodically, while Θ_k is selecting the next hop, it must also check the relay availability for those neighbors that are nearer than Θ_k to the addressed RSU (at most $\#\mathcal{N}_{\Theta_k}$ more checks of a boolean variable).

- Periodically, if Θ_k does not have any available next hop, it must check its own relay availability. In such case, lines 30 to 35 of Algorithm 1 must be executed ($\#\mathcal{N}_{\Theta_k}$ comparisons for the first step detailed in Section 5.1 and at most $(\#\mathcal{N}_{\Theta_k}) \cdot (\#\mathcal{N}_{\Theta_k} - 1)$ comparisons for the second step detailed in the same Section).

Given the capabilities of today devices, the complexity increase compared to GF can be considered negligible.

6. Centralized approach: greedy forwarding with virtual RSUs

The second proposed algorithm, GFVIR, has a preliminary centralized design phase, to be performed before the OBUs start using the routing algorithm. During the preliminary design phase, the position of local minima are identified and alternative paths are found. The hereafter defined attractive virtual roadside units (AVRSUs) and stopping virtual roadside units (SVRSUs) are then conveniently positioned per each (real) RSU and this information is provided to the OBUs. These virtual RSUs will participate to the routing process as detailed in the following; even if they are characterized by position and range, they are not real RSUs and do not imply any deployment with related costs. The addition of AVRSUs and SVRSUs only consists in new entries in the RSU database, managed by the location service (see Section 4).

A suitable choice of AVRSU and SVRSU positions helps the OBUs to avoid local minima. The role of attractive and stopping virtual RSUs will be better described in the following Subsection.

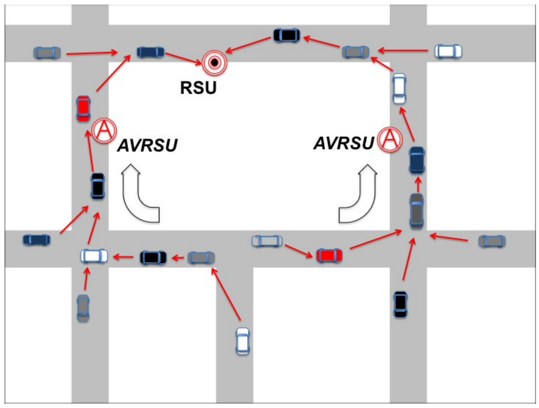
6.1. Attractive virtual road side units

Most local minima, like the one shown in Fig. 1(b), can be avoided forcing data flows along desired paths. To this aim, AVRSUs are placed in suitable positions to attract the traffic flows: OBUs will address them instead of the real RSU until the AVRSU proximity is reached. Then, the local minimum is overtaken and the real RSU can be addressed. As an example, in Fig. 3(a) the local minimum is avoided by opportunistically placing two AVRSUs.

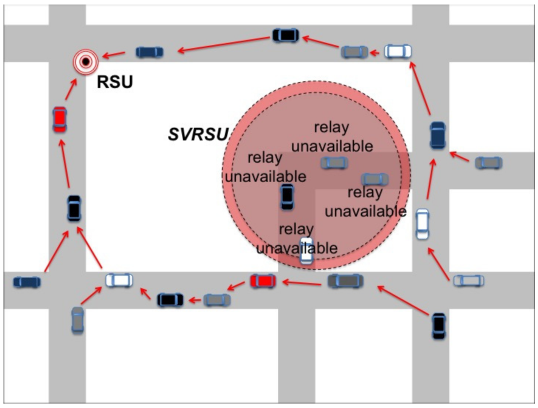
More specifically, in GFVIR the location service provides per each real RSU a list of AVRSUs. The generic AVRSU, $V_j^{(A)}$, is characterized by three parameters: 1) its position, 2) the reference real RSU, and 3) an *exclusion* distance, $\rho_{V_j^{(A)}}$. Denoting with Θ_k the generic OBU, with R_{Θ_k} the nearest RSU to Θ_k, and with $\mathcal{V}_{\Theta_k}^{(A)}$ the set of AVRSUs belonging to R_{Θ_k}, the following definition holds.

Definition 2. ***AVRSU availability.*** *An AVRSU, $V_j^{(A)} \in \mathcal{V}_{\Theta_k}^{(A)}$, is said to be* available *for Θ_k toward RSU R_{Θ_k} if:*

1. *$V_j^{(A)} - R_{\Theta_k}$ distance is less than $\Theta_k - R_{\Theta_k}$ distance (otherwise $V_j^{(A)}$ deviates data farther from the real R_{Θ_k});*
2. *$V_j^{(A)} - \Theta_k$ distance is less than $\Theta_k - R_{\Theta_k}$ distance (otherwise $V_j^{(A)}$ is farther from Θ_k than R_{Θ_k});*
3. *$V_j^{(A)} - \Theta_k$ distance is larger than the exclusion distance $\rho_{V_j^{(A)}}$ (the AVRSU is useful only if the OBU is far enough).*



(a) Example of AVRSUs.



(b) Example of SVRSU.

Figure 3: GFVIR. Examples of AVRSU and SVRSU deployment and use. Red solid arrows are used for the connections that will finally reach the RSU.

The available AVRSUs will be used by the OBUs as detailed in Section 6.3.

6.2. Stopping virtual road side units

In some cases, traffic flows cannot be simply deviated from local minima through AVRSUs. Observing the Example 2 shown in Fig. 1(c), no AVRSU can be effectively placed nearer to the RSU than the highlighted local minimum area. In such cases, SVRSUs are used. The generic SVRSU, denoted

Algorithm 2 GF with Virtual RSUs

```
procedure BY OBU $\Theta_k$

$\mathcal{R}$: set of real RSUs
$R_{\Theta_k}$: nearest real RSU to OBU $\Theta_k$
$\mathcal{V}^{(A)}_{R_{\Theta_k}}$ : set of AVRSUs referred to real RSU $R_{\Theta_k}$
$\mathcal{V}^{(S)}_{R_{\Theta_k}}$ : set of SVRSUs referred to real RSU $R_{\Theta_k}$
$A_{\Theta_k}$: addressed RSU for OBU $\Theta_k$
$\mathcal{N}_{\Theta_k}$: set of neighbors of OBU $\Theta_k$
$H_{\Theta_k}$: next hop for OBU $\Theta_k$
$\omega_X$: relay availability of OBU $X$
$T_B$: beacon interval

Every $T_B$:

    // Reset relay availability and next hop
    $\omega_{\Theta_k}$ := true
    $H_{\Theta_k}$ := null

    // The nearest (real) RSU is searched
    $R_{\Theta_k} := argmin_{R_r \in \mathcal{R}}\{d(\Theta_k, R_r)\}$

    // Check if the nearest RSU provides coverage
    if $\Theta_k$ is connected to $R_{\Theta_k}$ then
        $H_{\Theta_k} := R_{\Theta_k}$
    else
        // The addressed RSU is selected
        $A_{\Theta_k} := R_{\Theta_k}$ // Reset the addressed RSU
        $d_A := d(\Theta_k, R_{\Theta_k})$ // Reset the min. distance
        for all $V^{(A)}_w \in \mathcal{V}^{(A)}_{R_{\Theta_k}}$ do
            if $d(V^{(A)}_w, R_{\Theta_k}) < d(\Theta_k, R_{\Theta_k})$ then
                if $d(V^{(A)}_w, \Theta_k) < d(\Theta_k, R_{\Theta_k})$ then
                    if $d(V^{(A)}_w, \Theta_k) > \rho_{V^{(A)}_w}$ then
                        // $V^{(A)}_w$ is available
                        if $d(\Theta_k, V^{(A)}_w) < d_A$ then
                            $A_{\Theta_k} := V^{(A)}_w$
                            $d_A := d(\Theta_k, V^{(A)}_w)$
        // $A_{\Theta_k}$ is the addressed RSU
        // The next hop is searched in $\mathcal{N}_{\Theta_k}$
        $d_{min} := d(\Theta_k, A_{\Theta_k})$ // Reset the min. distance
        for all $N_w \in \mathcal{N}_{\Theta_k}$ do
            // Evaluate relay availability
            $\omega_{N_w}$ := true
            for all $V^{(S)}_y \in \mathcal{V}^{(S)}_k$ do
                if $d(\Theta_k, V^{(S)}_y) < \rho_{V^{(S)}_y}$ then
                    $\omega_{N_w}$ := false // Relay unavailable
                    break
            // Proceed only if $N_w$ is relay available
            if $\omega_{N_w}$ = true then
                if $d(N_w, \Theta_k) < d_{min}$ then
                    $H_{\Theta_k} := N_w$
                    $d_{min} := d(N_w, R_{\Theta_k})$

    // If $H_{\Theta_k} \neq$ null the next hop is addressed
    if $H_{\Theta_k} \neq$ null then
        Transmit data to $H_{\Theta_k}$ in the service channel
    else
        Store and carry data

    // In any case, send the beacon
    Send beacon in the control channel
```

by $V^{(S)}_j$, is again characterized by three parameters: 1) its position, 2) the reference real RSU, and 3) an *exclusion* distance, $\rho_{V^{(S)}_j}$. Whenever an OBU is covered by any of the SVRSUs, then the OBU is unavailable to act as relay for its neighbors. Thus, denoting with Θ_k the generic OBU, with R_{Θ_k} the nearest RSU to Θ_k, and with $\mathcal{V}^{(S)}_{\Theta_k}$ the set of SVRSUs belonging to R_{Θ_k}, the following definition holds.

Definition 3. ***GFVIR relay unavailability.*** *Θ_k is (GFVIR) relay unavailable if there is at least one $V^{(S)}_j \in \mathcal{V}^{(S)}_{\Theta_k} : d(\Theta_k, V^{(S)}_j) < \rho_{V^{(S)}_j}$.*

An OBU which is not relay unavailable is said *relay available.*

Note that, differently from GFAVR, in this case no advertisement of the relay availability is needed. Each OBU is, in fact, able to autonomously calculate the relay availability of all its neighbors.

6.3. The GFVIR protocol

Each OBU sends a normal beacon frame every T_B, assumed the same for all OBUs for simplicity.

Each OBU Θ_k which is not covered by an RSU performs the following algorithm to select the next hop.

1. Θ_k finds the nearest (in the Euclidean sense) RSU R_{Θ_k};
2. Θ_k identifies the set $\mathcal{V}^{(A)}_{\Theta_k}$ of AVRSUs and the set $\mathcal{V}^{(S)}_{\Theta_k}$ of SVRSUs referred to R_{Θ_k};
3. Θ_k finds the nearest available AVRSU in $\mathcal{V}^{(A)}_{\Theta_k}$, if any, and selects it as *addressed* RSU; if $\mathcal{V}^{(A)}_{\Theta_k}$ is empty or none of the AVRSUs in $\mathcal{V}^{(A)}_{\Theta_k}$ is available, then Θ_k selects R_{Θ_k} as *addressed* RSU;
4. Θ_k defines the set $\mathcal{N}^*_{\Theta_k}$ of the neighbor OBUs that are relay available (i.e., that are not covered by any SVRSU in $\mathcal{V}^{(S)}_{\Theta_k}$) AND closer to the *addressed* RSU;
5. If $\mathcal{N}^*_{\Theta_k}$ is empty, then no next relay is available for Θ_k. Otherwise, Θ_k assumes as next relay the OBU in $\mathcal{N}^*_{\Theta_k}$ which is the closest to R_{Θ_k}.

To follow possible variations in the topology, in our implementation we assume all vehicles repeat the algorithm before sending their beacon frame, which occurs every T_B seconds.

A pseudo code description of the algorithm is shown in Algorithm 2.

Table 2: Simulation parameters and output figures. (*) denotes values that are used when not otherwise specified.

Inputs

Symbol	Meaning	Assumed values
$\mathcal{E}$	Effective radiated power (EIRP)	23 dBm
$P_{r_{min}}$	Receiver sensitivity	-85 dBm
G_r	Antenna gain at the receiver	3 dB
γ_{min}	Threshold signal to interference plus noise ratio	10 dB
d_{tx}	Transmission range in the absence of obstacles and interferers	200 m (*)
B	Payload size of MAC frames	100 bytes
δ_{OBU}	Portion of vehicles equipped with the OBU	1 (*)
T_s	Period of acquisition from sensors at the OBU	10 s in Bologna 30 s in Cologne
λ	Data generation rate	$1/T_s$ packets/s
N_{MAX}	Buffer size	10000 (*)
N_{V2C}	Packets sent through V2C when N_{MAX} is reached	$0.2 \cdot N_{MAX}$
T_{out}	Maximum delivery delay, i.e. time deadline triggering V2C	∞ (*)

Outputs

Symbol	Meaning	Range
D_R	Rate of packets delivered to the RSU	$\in [0, 1]$
L	Average delay	≥ 0
N_{hops}	Average number of hops per generated packet	≥ 0

6.4. Complexity of GFVIR

Compared to GF, the GFVIR protocol implies a design phase to set the AVRSUs and SVRSUs with their parameters, an increase of the database used by the location service and the routing protocol to also include the AVRSUs and SVRSUs and a very small increase of complexity in the routing protocol performed by each OBU. No modification to the beacon messages is needed in this case. More specifically, with Θ_k denoting the generic OBU, R_{Θ_k} the nearest RSU to Θ_k, $\mathcal{V}^{(A)}_{\Theta_k}$ the set of AVRSUs referred to R_{Θ_k}, $\mathcal{V}^{(S)}_{\Theta_k}$ the set of SVRSUs referred to R_{Θ_k}, $\mathcal{N}_{\Theta_k}$ the set of neighbors of Θ_k, and $\#\mathcal{X}$ the cardinality of set $\mathcal{X}$, the following additional elements are required.

- AVRSUs and SVRSUs positions and exclusion distances must be defined. This operation is needed each time a real RSU is deployed and it should be repeated in the case of modifications to the traffic flows (such as if a new road is added).

- The RSU database used by the location service and the routing algorithm also includes the AVRSUs and SVRSUs.

- Periodically, when Θ_k has selected the nearest RSU R_{Θ_k}, it must also check the distance from the AVRSUs referred to R_{Θ_k}, throughout lines 27-36 of Algorithm 2 ($\#\mathcal{V}^{(A)}_k$ comparisons).

- When Θ_k has selected the addressed RSU, it must also check the relay availability for those neighbors that are nearer than Θ_k to the addressed RSU, throughout lines 39-51 of Algorithm 2 (at most $(\#\mathcal{N}_{\Theta_k}) \cdot (\#\mathcal{V}^{(S)}_k)$ comparisons).

Also in this case, given the capabilities of today devices, the complexity increase can be considered negligible.

7. Simulation tools and settings

Results are shown by means of simulations that take into account the joint effects of vehicular mobility and wireless communications. More specifically, the simulation platform for heterogeneous interworking networks (SHINE) [39, 40, 41] was used, which is a wireless network simulator designed and developed to reproduce the whole network architecture from the application to the physical layer. Realistic urban vehicular traces are used to reproduce the vehicle positions and movements.

A summary of the main input and output figures is given in Table 2. Hereafter, all the settings and observed outputs will be detailed.

7.1. WAVE/IEEE 802.11p simulations

OBUs are equipped with the WAVE technology [3]; WAVE defines, through the IEEE 1609 specifications, the communication system architecture

(a) Bologna. Area: 1.6 x 1.8 (2.88 km^2).

(b) Cologne. Area: 4.1 x 3.1 (12.71 km^2).

Figure 4: Road layouts, with the placement of the real RSU and the main local minima. The placement of AVRSU and SVRSU used by GFVIR is also shown. The size of symbols follows the real RSU transmission range (when $d_{tx} = 200$ m) or the AVRSU/SVRSU exclusion distance.

Table 3: Scenarios.

Scenario	**Area**	**Average n. of vehicles**
Bologna A ([4, 41], normal traffic)	2.88 km^2	455
Bologna B ([4, 41], heavy traffic)		670
Cologne ([43], 7:10-7:20 a.m.)	12.71 km^2	4280

and the complementary set of services and interfaces for vehicular scenarios; MAC and physical layer protocols are described by IEEE 802.11p.

As foreseen by the regulations of most Countries, multiple non overlapping channels of 10 MHz each, transmitted in the dedicated short range communications (DSRC) band around 5.9 GHz are assumed [42]. One of these channels is reserved for control purposes, where beacons are sent by both OBUs and RSUs at a frequency of 10 Hz, whereas a parallel service channel is assumed for the CSVN service.

Medium access control. The carrier sense multiple access with collision avoidance (CSMA/CA) MAC procedure foreseen by IEEE 802.11p is reproduced in details, with the sensing and random access procedures, with collisions and retransmissions, and also including hidden terminals, exposed terminals, and capture effect.

Channel Model. The following propagation model is assumed.

$$\mathrm{PL}(d) = \mathrm{PL}_0(1) + 10\beta \log_{10}(d) \tag{3}$$

where $\mathrm{PL}_0(1)$ is the free space path loss at 1 meter distance, β is the path loss exponent, and d is the distance in meters.

A threshold model is then assumed for the packet error rate, with a shadowing effect due to buildings: a transmission between two devices is possible only if the virtual line connecting them do not cross any building and the received power P_r is higher than the receiver sensitivity $P_{r_{min}}$; a transmission successfully completes if the average signal to noise and interference ratio (SINR) is higher than a threshold γ_{min}, otherwise an error (or a collision) occurs. This model is similar to the one adopted in previous works, such as [44] and [45], with the addition of the realistic effect of buildings, well motivated for example in [46].

Defining the *maximum transmission range* d_{tx} as the distance that corresponds to γ_{min} in the absence of obstacles and interference, in the following various values for β (between 2.42 and 3.72) will be considered, corresponding to a different maximum transmission range d_{tx} (between 50 and 300 meters). $d_{tx} = 200$ m is used when not differently specified, corresponding to $\beta = 2.75$, coherently with measurements shown in [47].

7.2. Scenarios and application settings

Two cities with realistic vehicular traffic are considered as case studies: 1) a 2.88 km^2 central portion of the Italian city of Bologna (as detailed in [4, 41]), and 2) a 12.71 km^2 central portion of the German city of Cologne (a portion of the scenario described in [43]). Two values for the vehicle density are considered in the Bologna case, as summarized in Table 3. In all cases, a portion δ_{OBU} of the vehicles is equipped with the OBU (with $\delta_{OBU} = 1$ where not differently specified). The three scenarios have different amounts of vehicles and different distributions; the use as case studies of two cities and variable densities allows us to prove the general effectiveness of the proposed protocols.

A single RSU is placed in front of the main railway station in both cities. When GFVIR is considered, 3 AVRSUs and 4 SVRSUs have been placed in the Bologna scenario, whereas 14 AVRSUs and 3 SVRSUs have been placed in the Cologne scenario. The AVRSU and SVRSU placements have been heuristically optimized, following the position of the main local minima in both scenarios. The road layouts, the main local minima, and the real and virtual RSU placements in Bologna and Cologne are shown in Fig. 4.

Concerning the application, all OBUs acquire data from sensors and generate a new packet of $B = 100$ bytes every T_s seconds, that is, with a data generation rate $\lambda = 1/T_s$ p/s (we will use p to denote packets for brevity). Packets are stored in the OBU transmitter queue until the RSU is reached, a given maximum number of packets N_{MAX} is buffered, or a time out is triggered. In particular, the number of packets in the queue and the timestamp of the oldest packet are periodically checked. When N_{MAX} packets are buffered, a portion $N_{TX} = 0.2 \cdot N_{MAX}$ is sent to the control center through V2C to avoid data loss. If the oldest packet was generated more than T_{out} seconds earlier, then all packets are sent through V2C.

7.3. Output Figures

The system performance is evaluated in terms of the following metrics:

- D_R, which is the ratio of packets delivered to the control center through the RSU (i.e., using V2V and V2R),

$$D_R \triangleq \frac{\varphi_{RSU}}{\varphi_{gen}} \tag{4}$$

  where φ_{gen} is the overall number of packets generated, and φ_{RSU} is the number of packets transferred to the RSU using V2V and V2R communications;

- L, which is the average delay of delivered packets;

- N_{hops}, which is the average number of hops per packet,

$$N_{hops} \triangleq \frac{\varphi_{RSU} + \varphi_{V2V}}{\varphi_{gen}} \tag{5}$$

  where φ_{V2V} is the number of successful V2V transmissions.

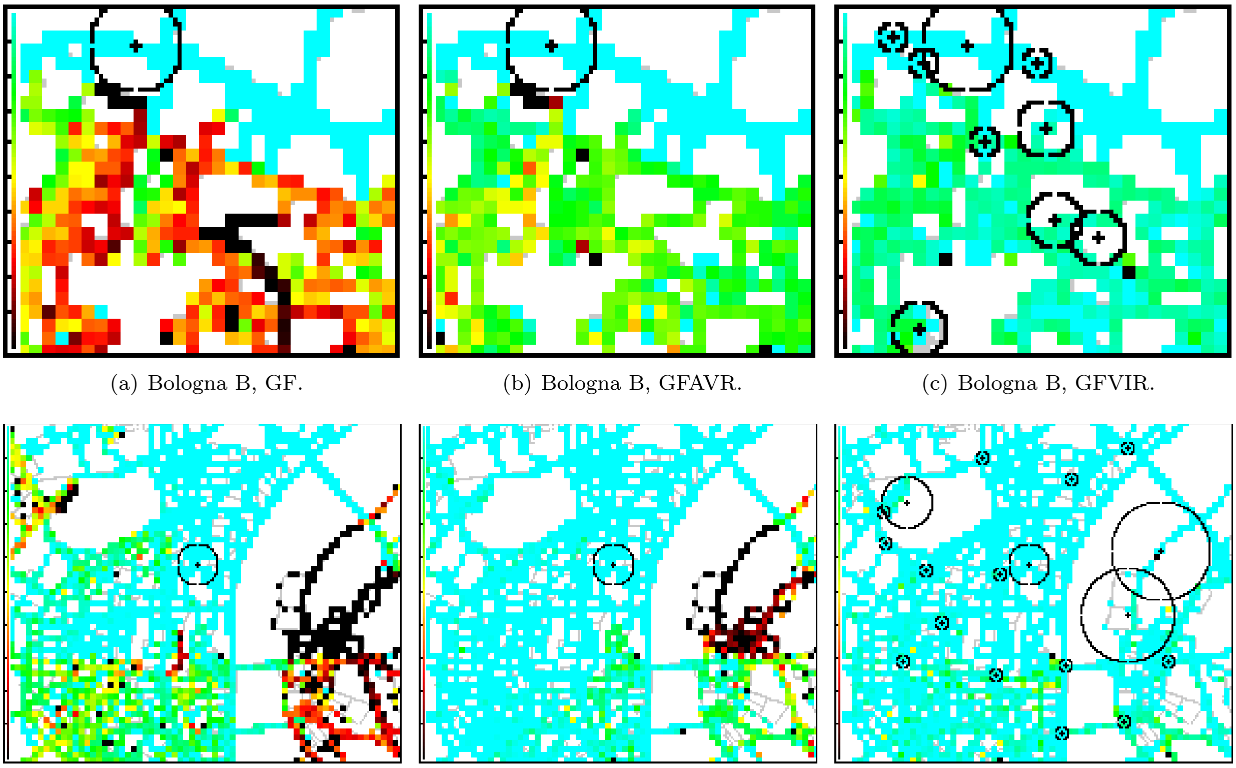
8. Numerical results

The performance of GFAVR and GFVIR is shown through Figs. 5-11. The 90% t-based confidence interval is presented in some curves, whereas in the others it was extremely small and was removed for the sake of readability.

The effectiveness of the proposed algorithms is shown, in Fig. 5, in terms of D_R varying δ_{OBU}. The first noticeable conclusion is that both the proposed algorithms show a higher performance compared to GF, for moderate and large percentages of vehicles equipped with OBUs. In scenarios with small node density and small local minimum areas (e.g., Bologna A with $\delta_{OBU} \leqslant 0.5$), the three algorithms tend to behave similarly. In such case, the network of nodes is sparse and often nodes have few neighbors. For this reason, OBUs that travel in a small local minimum area have high probability to store and carry the packets outside that area, and the local minima problem rarely arises.

Results also confirm that the basic GF routing algorithm provides a good D_R, with more than 60% packets delivered to the RSU in all scenarios, even with $\delta_{OBU} = 0.25$. Still focusing on GF, it is also interesting to note that D_R increases with an increase of δ_{OBU}, thanks to the higher density. Once a maximum value is reached, however, D_R tends to reduce due to the higher impact of local minima.

Fig. 5 also shows that both GFAVR and GFVIR provide a relevant improvement in terms of D_R when $\delta_{OBU} = 1$, with an increase that ranges from 10% to 16% for the former protocol, and from 12% to 24% for the latter one, according to the considered scenarios. As expected, thanks to the preliminary design phase, GFVIR allows a higher improvement compared to GFAVR. On the other hand, the



(a) Bologna B, GF. (b) Bologna B, GFAVR. (c) Bologna B, GFVIR.

(d) Cologne, GF. (e) Cologne, GFAVR. (f) Cologne, GFVIR.

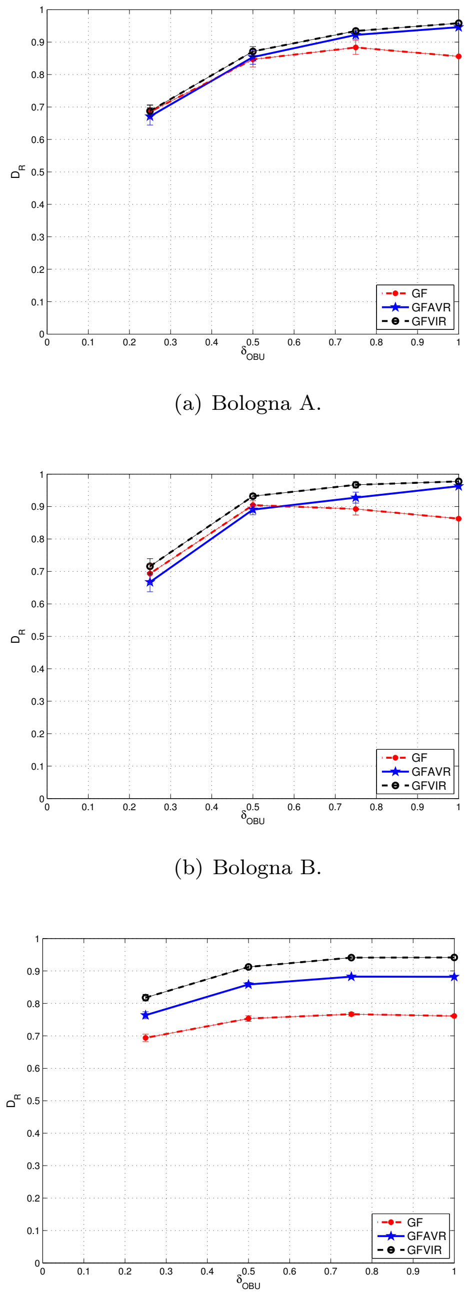
Figure 8: Probability that the data generated in each position reached the control center through V2V and V2R (brighter) or through V2C (darker). Results refer to $d_{max} = 200$ m, $N_{MAX} = 500$, and $T_{out} \to \infty$. The real and virtual RSUs are highlighted in black with their transmission range or exclusion distance.

design phase of GFVIR is specific for the addressed scenario, while GFAVR is fully distributed and independent from the scenario.

In Figs. 6 and 7, D_R is shown varying N_{MAX} and T_{out}, respectively. In general, large values of N_{MAX} or T_{out} are expected to increase the probability that modifications to the topology due the vehicle mobility create new paths toward the RSU. This is indeed observable both in Fig. 6 and in Fig. 7, where D_R grows increasing N_{MAX} or T_{out}. Note, however, that in all cases a maximum is reached, and increasing N_{MAX} to more than 3000 or T_{out} to more than 120 s has a negligible impact.

Focusing on the case with the highest gap in terms of D_R (i.e., $N_{MAX} = 500$ and $T_{out} \to \infty$), Fig. 8 highlights the effect of local minima on the distribution of data loss. More specifically, Fig. 8 shows, for Bologna B and Cologne, the rate of packets generated in each position of the scenario that are sent through the RSU instead of through V2C; a lighter color is used for a higher rate of packets reaching the RSU (light blue means 100% reach the RSU, black means 100% packets are sent through V2C). The impact of GF, GFAVR, and GFVIR is shown in the subfigures. As observable in Figs. 8(a) and 8(d), in the case of GF the local minima prevent most packets generated in some areas to reach the RSU. This effect is reduced by GFAVR (Figs. 8(b) and 8(e)) and almost eliminated by GFVIR (Figs. 8(c) and 8(f)). Compared to GF, GFAVR leads to an increase of D_R of 28% in Bologna B and 24% in Cologne, whereas GFVIR allows an increase of D_R of 46% in Bologna B and 32% in Cologne.

The results shown in Fig. 8 also remark the effect of data loss if no V2C was foreseen. Besides the data loss, which is a flaw that some applications might tolerate, the main drawback is that losses are not evenly distributed, but concentrated in specific areas. Under these conditions, the CSVN application would not be able to provide information about some specific areas, irrespective to the amount of

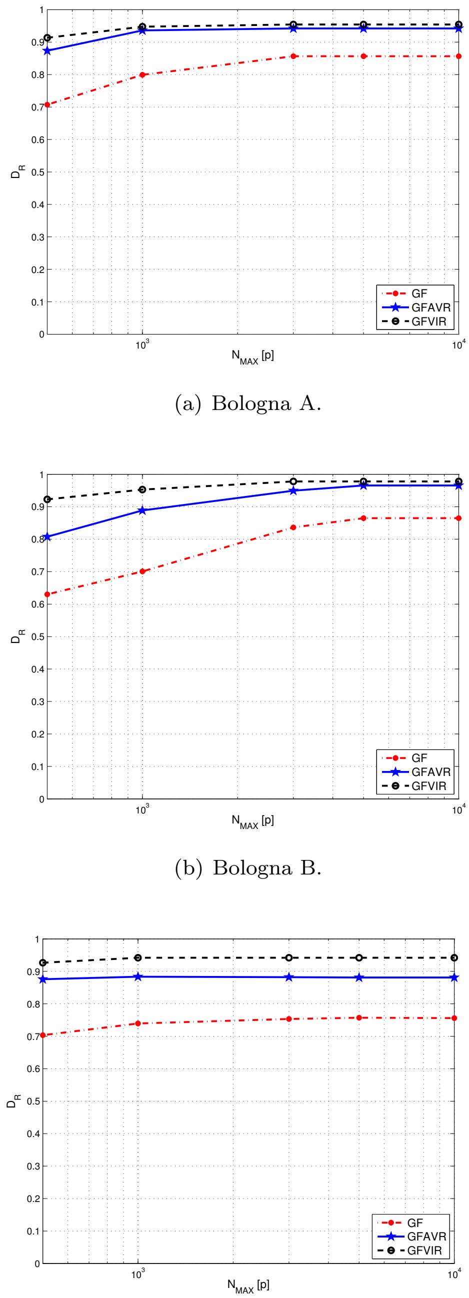


(a) Bologna A.

(b) Bologna B.

(c) Cologne.

Figure 5: Delivery rate vs. portion of equipped vehicles.



(a) Bologna A.

(b) Bologna B.

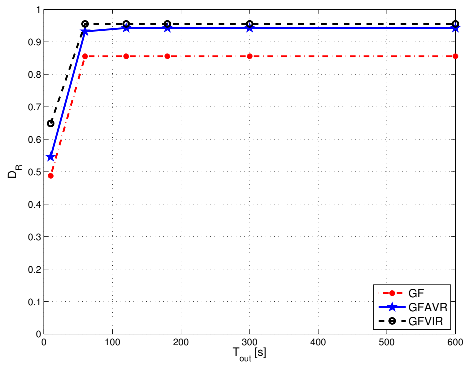
(c) Cologne.

Figure 6: Delivery rate vs. buffer size.

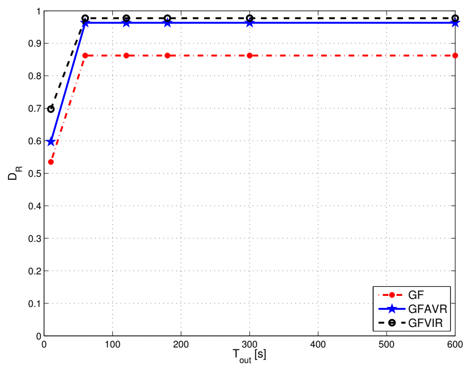
collected data.

Varying d_{tx}, further results are shown in Figs. 9, 10, and 11, in terms of D_R, L and N_{hops}, respectively. Focusing on Fig. 9, similar conclusions as those provided can be drawn. In this case, a lower effectiveness of GFAVR is observable when a small d_{tx} is assumed. In such case, the OBUs have less neighbors, thus they have less information to correctly determine their relay availability.

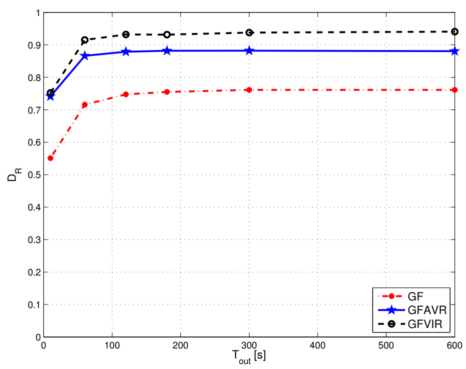
The effect on L is shown in Fig. 10. Also in terms



(a) Bologna A.

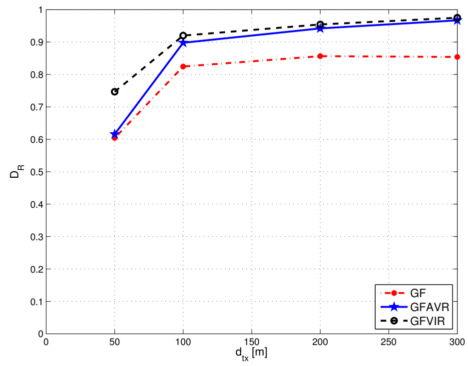


(b) Bologna B.

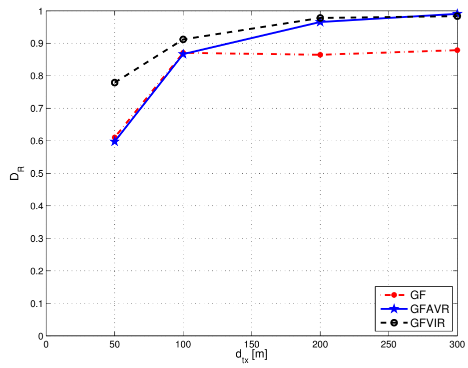


(c) Cologne.

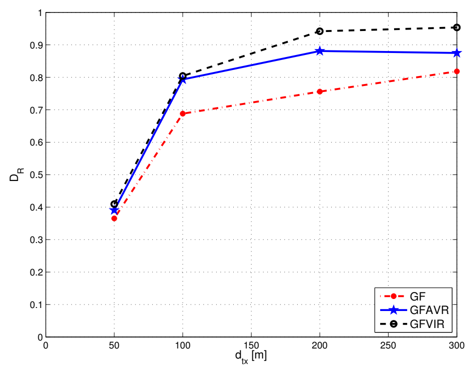
Figure 7: Delivery rate vs. time out.



(a) Bologna A.



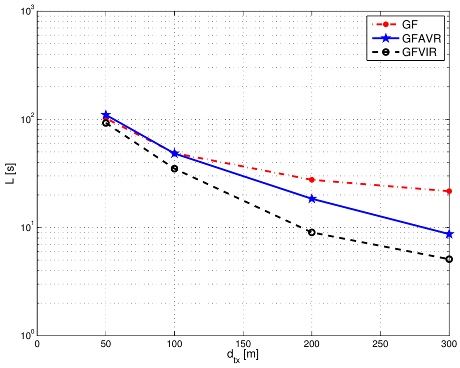
(b) Bologna B.



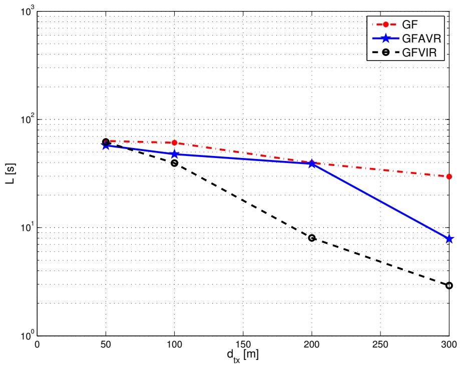
(c) Cologne.

Figure 9: Delivery rate vs. transmission range.

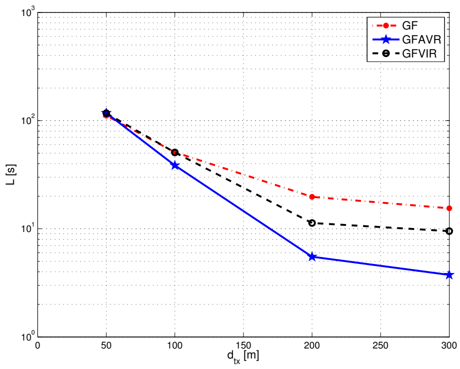
of delay, GFAVR and GFVIR are shown to outperform GF. Although GFVIR makes, in general, packets traveling longer paths toward the RSU, and although both algorithms increase the probability that packets remain stored on board of OBUs due to the absence of a next hop, they still allow lower delay than GF in most cases. The longer paths and the holding delay, in fact, are balanced by a lower probability that part of packets are blocked for some time in the local minima.



(a) Bologna A.

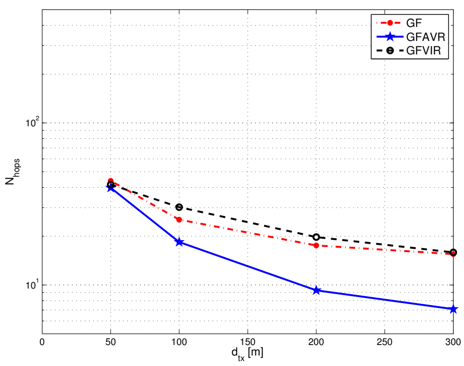


(b) Bologna B.

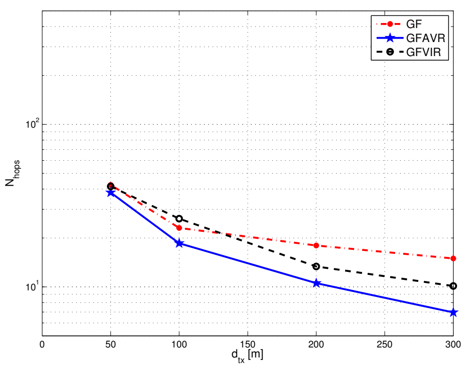


(c) Cologne.

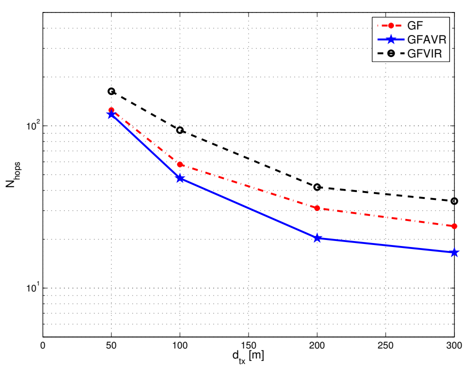
Figure 10: Average delivery delay vs. transmission range.



(a) Bologna A.



(b) Bologna B.



(c) Cologne.

Figure 11: Average number of hops vs. transmission range.

Finally, results are shown in terms of N_{hops} in Fig. 11. In this case, several conflicting effects contribute to the results: 1) The use of AVRSUs normally implies longer paths to avoid local minima (in terms of number of hops), thus affecting GFVIR with an higher value of N_{hops}; 2) In the areas where local minima are located, packets tend to be passed by one vehicle to another, until N_{MAX} or T_{out} are reached; this tends to increase N_{hops} in GF; 3) A packet, which is sent through the cellular link, also contributes to this metric, sometimes even with a number of hops equal to 0 (if it is directly sent through the cellular link). Looking at the results shown in Fig. 11, GFAVR always provides the lowest N_{hops}, whereas GF or GFVIR cause the highest value, depending on the scenario. However, note that the number of hops and the average delivery delay are not strictly proportional to each other, as observable comparing the average number of hops N_{hops} of Fig. 11 with the delivery delay L of Fig. 10. This is due to the store and carry ability of OBUs, that impacts on delay and not on the number of hops.

Summarizing the results shown in Figs. 5-11, GFAVR provided up to 28% higher D_R compared to GF, with a lower average delivery delay and a lower average number of hops. GFVIR provided up to 46% higher D_R compared to GF, with a lower average delivery delay and similar or slightly higher average number of hops. Both the algorithms tend to provide similar performance than GF if the density of nodes is very low and the local minimum areas are small.

9. Conclusion

In this paper, two novel routing protocols, GFAVR and GFVIR, have been proposed to overcome the local minima problem in VANETs, which arises when a GF approach is adopted to address fixed RSUs. The former algorithm is fully distributed, does not need any a priori knowledge of the scenario, and adds a single overhead bit. The latter requires a preliminary design phase to individuate the main local minima and alternative paths in the addressed scenario and it needs an increase of the RSU database, but does not imply any additional signaling overhead. Whereas GFAVR is simpler to implement and independent from the specific scenario, GFVIR provides better performance in most cases. Results, obtained through extensive simulations in realistic urban scenarios demonstrated that both algorithms significantly improve the delivery rate and reduce the average delivery delay compared to GF, proving they are suitable choices for network routing in CSVNs.

References

[1] A. Bazzi, B. Masini, A. Zanella, G. Pasolini, Virtual road side units for geo-routing in VANETs, in: International Conference on Connected Vehicles & Expo (ICCVE 2014), 2014.

[2] NHSTA web page, accessed on June 2015. URL http://www.nhtsa.gov

[3] R. Uzcategui, G. Acosta-Marum, WAVE: A tutorial, Communications Magazine, IEEE 47 (5) (2009) 126–133. doi:10.1109/MCOM.2009.4939288.

[4] A. Bazzi, B. Masini, G. Pasolini, V2V and V2R for cellular resources saving in vehicular applications, in: Wireless Communications and Networking Conference (WCNC), 2012 IEEE, 2012, pp. 3199–3203. doi:10.1109/WCNC.2012.6214358.

[5] R. Ganti, F. Ye, H. Lei, Mobile crowdsensing: current state and future challenges, Communications Magazine, IEEE 49 (11) (2011) 32–39. doi:10.1109/MCOM.2011.6069707.

[6] H. Ma, D. Zhao, P. Yuan, Opportunities in mobile crowd sensing, Communications Magazine, IEEE 52 (8) (2014) 29–35. doi:10.1109/MCOM.2014.6871666.

[7] X. Yu, H. Zhao, L. Zhang, S. Wu, B. Krishnamachari, V. Li, Cooperative sensing and compression in vehicular sensor networks for urban monitoring, in: Communications (ICC), 2010 IEEE International Conference on, 2010, pp. 1–5. doi:10.1109/ICC.2010.5502562.

[8] A. R. Al-Ali, I. Zualkernan, F. Aloul, A mobile GPRS-sensors array for air pollution monitoring, Sensors Journal, IEEE 10 (10) (2010) 1666–1671. doi:10.1109/JSEN.2010.2045890.

[9] A. Bazzi, B. Masini, O. Andrisano, On the frequent acquisition of small data through RACH in UMTS for ITS applications, Vehicular Technology, IEEE Transactions on 60 (7) (2011) 2914–2926. doi:10.1109/TVT.2011.2160211.

[10] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, M. Mauve, Data aggregation and roadside unit placement for a VANET traffic information system, in: Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking, VANET '08, ACM, New York, NY, USA, 2008, pp. 58–65. doi:10.1145/1410043.1410054.

[11] B. Aslam, F. Amjad, C. Zou, Optimal roadside units placement in urban areas for vehicular networks, in: Computers and Communications (ISCC), 2012 IEEE Symposium on, 2012, pp. 000423–000429. doi:10.1109/ISCC.2012.6249333.

[12] N. Benamar, K. D. Singh, M. Benamar, D. E. Ouadghiri, J.-M. Bonnin, Routing protocols in vehicular delay tolerant networks: A comprehensive survey, Computer Communications 48 (0) (2014) 141 – 158. doi:10.1016/j.comcom.2014.03.024.

[13] Z. Taysi, A. Yavuz, Routing protocols for geonet: A survey, Intelligent Transportation Systems, IEEE Transactions on 13 (2) (2012) 939–954. doi:10.1109/TITS.2012.2183637.

[14] M. Mauve, J. Widmer, H. Hartenstein, A survey on position-based routing in mobile ad hoc networks, Network, IEEE 15 (6) (2001) 30–39. doi:10.1109/65.967595.

[15] B. Karp, H. T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom '00, ACM, New York, NY, USA, 2000, pp. 243–254. doi:10.1145/345910.345953.

[16] C. Lochert, M. Mauve, H. Füssler, H. Hartenstein, Geographic routing in city scenarios, SIGMOBILE Mob. Comput. Commun. Rev. 9 (1) (2005) 69–72. doi:10.1145/1055959.1055970.

[17] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, A. Corradi, Efficient data harvesting in mobile sensor platforms, in: Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on, 2006, pp. 5 pp.–356. doi:10.1109/PERCOMW.2006.47.

[18] G. Comert, M. Cetin, Queue length estimation from probe vehicle location and the impacts of sample size, European Journal of Operational Research 197 (1) (2009) 196 – 202. doi:10.1016/j.ejor.2008.06.024.

[19] R. Stanica, M. Fiore, F. Malandrino, Offloading floating car data, in: World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a, 2013, pp. 1–9. doi:10.1109/WoWMoM.2013.6583391.

[20] U. Lee, M. Gerla, A survey of urban vehicular sensing platforms, Computer Networks 54 (4) (2010) 527 – 544. doi:10.1016/j.comnet.2009.07.011.

[21] B. Masini, L. Zuliani, O. Andrisano, On the effectiveness of a GPRS based intelligent transportation system in a realistic scenario, in: Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd, Vol. 6, 2006, pp. 2997–3001. doi:10.1109/VETECS.2006.1683418.

[22] I. Leontiadis, G. Marfia, D. Mack, G. Pau, C. Mascolo, M. Gerla, On the effectiveness of an opportunistic traffic management system for vehicular networks, Intelligent Transportation Systems, IEEE Transactions on 12 (4) (2011) 1537–1548. doi:10.1109/TITS.2011.2161469.

[23] C. de Fabritiis, R. Ragona, G. Valenti, Traffic estimation and prediction based on real time floating car data, in: Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on, 2008, pp. 197–203. doi:10.1109/ITSC.2008.4732534.

[24] A. Skordylis, N. Trigoni, Efficient data propagation in traffic-monitoring vehicular networks, Intelligent Transportation Systems, IEEE Transactions on 12 (3) (2011) 680–694. doi:10.1109/TITS.2011.2159857.

[25] A. Bazzi, B. M. Masini, A. Zanella, G. Pasolini, {IEEE} 802.11p for cellular offloading in vehicular sensor networks, Computer Communications 60 (0) (2015) 97 – 108. doi:10.1016/j.comcom.2015.01.012.

[26] F. Li, Y. Wang, Routing in vehicular ad hoc networks: A survey, Vehicular Technology Magazine, IEEE 2 (2) (2007) 12–22. doi:10.1109/MVT.2007.912927.

[27] V. Naumov, T. Gross, Connectivity-aware routing (CAR) in vehicular ad-hoc networks, in: INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, 2007, pp. 1919–1927. doi:10.1109/INFCOM.2007.223.

[28] J. Haerri, F. Filali, C. Bonnet, Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns, in: Proc. of 5th IFIP Mediterranean Ad-Hoc Networking Workshop (Med-Hoc-Net-2006), Lipari, Italy, 2006.

[29] A. Vahdat, D. Becker, et al., Epidemic routing for partially connected ad hoc networks, Tech. rep., Technical Report CS-200006, Duke University (2000).

[30] T. Spyropoulos, K. Psounis, C. S. Raghavendra, Spray and wait: An efficient routing scheme for intermittently connected mobile networks, in: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking, WDTN '05, ACM, New York, NY, USA, 2005, pp. 252–259. doi:10.1145/1080139.1080143.

[31] Y. Xiang, Z. Liu, R. Liu, W. Sun, W. Wang, GeoSVR: A map-based stateless VANET routing, Ad Hoc Networks 11 (7) (2013) 2125–2135, doi:10.1016/j.adhoc.2012.02.015.

[32] K. Mershad, H. Artail, M. Gerla, ROAMER: Roadside units as message routers in VANETs, Ad Hoc Networks 10 (3) (2012) 479 – 496, doi:10.1016/j.adhoc.2011.09.001.

[33] S. Ahmed, S. S. Kanere, SKVR: Scalable knowledge-based routing architecture for public transport networks, in: Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks, VANET '06, ACM, New York, NY, USA, 2006, pp. 92–93, doi:10.1145/1161064.1161082.

[34] Q. Yuan, I. Cardei, J. Wu, An efficient prediction-based routing in disruption-tolerant networks, Parallel and Distributed Systems, IEEE Transactions on 23 (1) (2012) 19–31. doi:10.1109/TPDS.2011.140.

[35] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, M. Mauve, A routing strategy for vehicular ad hoc networks in city environments, in: Intelligent Vehicles Symposium, 2003. Proceedings. IEEE, 2003, pp. 156–161. doi:10.1109/IVS.2003.1212901.

[36] T. Camp, J. Boleng, L. Wilcox, Location information services in mobile ad hoc networks, in: Communications, 2002. ICC 2002. IEEE International Conference on, Vol. 5, 2002, pp. 3318–3324 vol.5. doi:10.1109/ICC.2002.997446.

[37] J. Bernsen, D. Manivannan, Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification, Pervasive and Mobile Computing 5 (1) (2009) 1 – 18. doi:10.1016/j.pmcj.2008.09.001.

[38] K. Lee, J. Haerri, U. Lee, M. Gerla, Enhanced perimeter routing for geographic forwarding protocols in urban vehicular scenarios, in: Globecom Workshops, 2007 IEEE, 2007, pp. 1–10. doi:10.1109/GLOCOMW.2007.4437832.

[39] A. Bazzi, G. Pasolini, C. Gambetti, SHINE: Simulation platform for heterogeneous interworking networks, in: Communications, 2006. ICC '06. IEEE International Conference on, Vol. 12, 2006, pp. 5534–5539. doi:10.1109/ICC.2006.255543.

[40] A. Toppan, A. Bazzi, P. Toppan, B. Masini, O. Andrisano, Architecture of a simulation platform for the smart navigation service investigation, in: Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on, 2010, pp. 548–554. doi:10.1109/WIMOB.2010.5645014.

[41] SHINE web page, accessed on June 2015.
URL http://www.wcsg.ieiit.cnr.it/people/bazzi/SHINE.html

[42] C. Campolo, A. Molinaro, Multichannel communications in vehicular ad hoc networks: a survey, Communications Magazine, IEEE 51 (5) (2013) 158–169. doi:10.1109/MCOM.2013.6515061.

[43] S. Uppoor, O. Trullols-Cruces, M. Fiore, J. Barcelo-Ordinas, Generation and analysis of a large-scale urban vehicular mobility dataset, Mobile Computing, IEEE Transactions on PP (99) (2013) 1–1. doi:10.1109/TMC.2013.27.

[44] A. Benslimane, S. Barghi, C. Assi, An efficient routing protocol for connecting vehicular networks to the internet, Pervasive and Mobile Computing 7 (1) (2011) 98 – 113. doi:10.1016/j.pmcj.2010.09.002.

[45] J.-J. Chang, Y.-H. Li, W. Liao, I.-C. Chang, Intersection-based routing for urban vehicular communications with traffic-light considerations, Wireless Communications, IEEE 19 (1) (2012) 82–88. doi:

10.1109/MWC.2012.6155880.

[46] F. Martinez, C.-K. Toh, J.-C. Cano, C. Calafate, P. Manzoni, Realistic radio propagation models (RPMs) for VANET simulations, in: Wireless Communications and Networking Conf. WCNC. IEEE, 2009, pp. 1 –6. doi:10.1109/WCNC.2009.4917932.

[47] L. Cheng, B. Henty, D. Stancil, F. Bai, P. Mudalige, Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz dedicated short range communication (DSRC) frequency band, IEEE J. Sel. Areas Commun. 25 (8) (2007) 1501 –1516. doi:10.1109/JSAC.2007.071002.